INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **34**: 167–185

Improvements to compressible Euler methods for low-Mach number flows

Murat Sabanca[a,*,1], Gunther Brenner[a] and Nafiz Alemdaroğlu[b,2]

[a] *Department of Fluid Mechanics (LSTM), University of Erlangen-Nuremberg, Cauerstrasse 4, Erlangen, Germany*
[b] *Department of Aeronautical Engineering, Middle East Technical University, Inonu Bulvari, Ankara, Turkey*

SUMMARY

In the present study improvements to numerical algorithms for the solution of the compressible Euler equations at low Mach numbers are investigated. To solve flow problems for a wide range of Mach numbers, from the incompressible limit to supersonic speeds, preconditioning techniques are frequently employed. On the other hand, one can achieve the same aim by using a suitably modified acoustic damping method. The solution algorithm presently under consideration is based on Roe's approximate Riemann solver [Roe PL. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics* 1981; **43**: 357–372] for non-structured meshes. The numerical flux functions are modified by using Turkel's preconditioning technique proposed by Viozat [Implicit upwind schemes for low Mach number compressible flows. INRIA, Rapport de Recherche No. 3084, January 1997] for compressible Euler equations and by using a modified acoustic damping of the stabilization term proposed in the present study. These methods allow the compressible Euler equations at low-Mach number flows to be solved, and they are consistent in time. The efficiency and accuracy of the proposed modifications have been assessed by comparison with experimental data and other numerical results in the literature. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: acoustic damping; Euler; low Mach number; preconditioning

1. INTRODUCTION

In recent years, increasing interest has been focused on the investigation of numerical methods for the simulation of flows at low Mach numbers. These investigations are mainly derived by the fact that many applications of technological interest occur in the transitional regime between incompressible and compressible flows. For these applications, an 'all Mach number capability' of the simulation code is required. Several numerical techniques addressing this

* Correspondence to: Department of Fluid Mechanics (LSTM), University of Erlangen-Nuremberg, Cauerstrasse 4, D-91058, Erlangen, Germany.
[1] E-mail: murat@lstm.uni-erlangen.de
[2] E-mail: nafiz@metu.edu.tr

Received 4 February 1999
Revised 19 October 1999

problem have been proposed. These are derived either from traditional incompressible methods or from the modifications of compressible methods. Incompressible methods that use pressure as the primary variable and a compressible form of the SIMPLE algorithm have been proposed in References [1,2]. A more rigorous treatment of the problem is the asymptotic analysis using multiple pressure variables [3,4], where the effect of acoustics on the flow field is treated via a Taylor expansion of the pressure in terms of Mach number. Various methods have been proposed to perform low-Mach number calculations with compressible codes [5–10]. Here, two techniques have been investigated to modify the strength of the acoustic waves and thus to improve convergence to a steady state. On the other hand, the methods belong to the *artificial compressibility* approach, applying a preconditioning operator to the time derivative, in which an originally mixed elliptic/hyperbolic system becomes a hyperbolic system of equations. This method is inconsistent in time. Thus, it is not suitable for unsteady problems. A variety of preconditioning strategies has been developed, suitable also for unsteady problems, as has been shown in Reference [11].

In the present investigation, a modified acoustic damping method is developed for compressible Euler equations. It is also suitable for unsteady problems. Moreover, it gives the same accuracy as unsteady preconditioning strategies when it is suitably modified. The implementation of this method is easier as well. In reality, both techniques damp the acoustic waves artificially. On the other hand, the use of numerical methods to solve partial differential equations (PDEs) introduces an approximation that can change the form of the basic differential equations themselves. Since these approximations are not precisely the same as the original equations, they can simulate physical phenomena like diffusion, dissipation, boundary layers, etc., in ways that are not exactly the same as an exact solution to the basic PDE. These differences are usually referred to as truncation errors, which are often identified with particular physical phenomena on which they have a strong effect. There is nothing wrong with identifying an error with a physical process as long as the error is kept small in some engineering sense. In the present study, these errors are referred to as dissipation errors. The concept of acoustic damping is applied to the unsteady compressible Euler equations with a modified formulation and without changing the dissipative properties of the flow. The obtained solutions are compared with the experimental results as well as with the numerical solutions of Viozat [11], who obtained them by changing the numerical flux function in a time-consistent manner, as is done in the present investigation. During this investigation, a finite volume, two-dimensional, unstructured Euler solver (SUN), both for external and internal flows, was developed by Sabanca [12] and validated for various test cases.

2. GOVERNING EQUATIONS

The equations of motion are the Euler equations, which govern the motion of an inviscid, compressible, unsteady fluid. They can be written in conservative differential form as follows:

$$\frac{\partial Q}{\partial t} + \vec{\nabla} \cdot \vec{F} = 0 \tag{1}$$

where Q is the vector of the conservative variables and $\vec{F} = (F, G)$ are the components of the inviscid fluxes, defined as

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix}, \qquad F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uh \end{bmatrix}, \qquad G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vh \end{bmatrix} \tag{2}$$

The relations for energy, enthalpy and pressure close the system

$$e = \frac{p}{(\gamma - 1)\rho} + \frac{1}{2}(u^2 + v^2), \qquad h = e + \frac{p}{\rho} \tag{3}$$

3. NUMERICAL METHOD

The integral conservation laws are written for a control volume Ω after applying Gauss's theorem as

$$\int_\Omega \frac{\partial Q}{\partial t}\, \mathrm{d}\Omega + \oint_{\partial\Omega} \vec{F} \cdot \hat{n}\, \mathrm{d}\sigma = 0 \tag{4}$$

where $\vec{n}$ is the outward-pointing unit normal vector to the control volume. The control volumes for the present cell vertex finite volume approximation of Equation (4) are chosen to be the barycentric cells around each node, as is shown in Figure 1. The upwinding direction in the spatial discretization is chosen to be the control volume's face normal. For the present unstructured formulations, these normals can be defined as follows: considering Figure 2, let

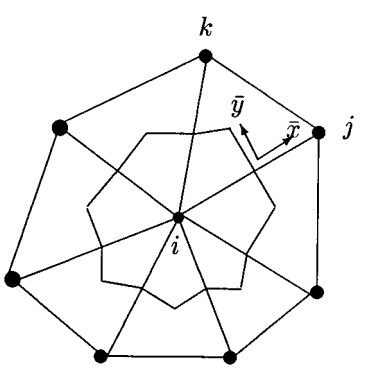

Figure 1. Control volumes are the barycentric cells around each node.

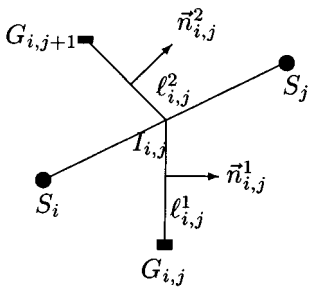

Figure 2. Cell face normals.

G_{ij} and G_{ij+1} be the centroids of two neighbouring triangles and let I_{ij} be the mid-point of the common edge of these two neighbouring triangles. Then, directed lengths from the dual mesh are defined by the corresponding normals and the lengths of the sections $\hat{n}^1_{ij} = \vec{n}^1_{ij}\ell^1_{ij}$ and $\hat{n}^2_{ij} = \vec{n}^2_{ij}\ell^2_{ij}$. An averaged directed length is then defined as $\hat{n}_{ij} = \hat{n}^1_{ij} + \hat{n}^2_{ij}$. With this notation, the second term in Equation (4) can be approximated as follows:

$$\oint_{\partial\Omega} \vec{F}\cdot\hat{n}\,\mathrm{d}\sigma \approx \sum_{j\in K(i)} \vec{F}_{ij}\cdot\hat{n}_{ij} \tag{5}$$

where $j\in K(i)$ denotes the set of all the neighbouring nodes of i and $\vec{F}_{ij}$ is some approximation of the convective flux computed on the edge between two adjacent control volumes.

3.1. Preconditioning of the flux function

The spatial discretization is based on a flux difference splitting nodal point scheme, which is modified to handle low-Mach number flows. The computation of the convective terms in Equation (4) is decomposed among the cell faces of the control volumes. The upwinding is introduced in the computation of this term through the use of numerical flux function Φ. Two kinds of numerical flux functions are used. The first one is proposed by Viozat [11]

$$\Phi_F(Q_\mathrm{L}, Q_\mathrm{R}, \hat{n}) = \frac{F(Q_\mathrm{L}, \hat{n}) + F(Q_\mathrm{R}, \hat{n})}{2} - \frac{1}{2} P_c^{-1} |P_c A_c \hat{n}_x + P_c B_c \hat{n}_y| (Q_\mathrm{R} - Q_\mathrm{L}) \tag{6}$$

For the detailed derivation see References [11,12]. Here, Q_L and Q_R are the conservative variables on the left and right states of the edge under consideration. A_c and B_c are the Jacobian matrices in conservative variables. P_c is the preconditioning matrix proposed by Turkel [9] for incompressible flows

$$P_c = I_{4\times 4} + (\beta^2 - 1)\frac{\gamma - 1}{a^2}\begin{bmatrix} \frac{q^2}{2} & -u & -v & 1 \\ \frac{q^2}{2}u & -u^2 & -vu & u \\ \frac{q^2}{2}v & -uv & -v^2 & v \\ \frac{q^2}{2}H & -uH & -vH & H \end{bmatrix} \tag{7}$$

The numerical flux function is written in a form that allows the comparison with central schemes in that the first term can be thought of as a central discretization term and the second term can be thought of as a stabilization or artificial viscosity term. From a diagonalizing matrix, the stabilization term can be obtained from its eigenvalues as follows:

$$P_c^{-1}\left|P_c A_c \hat{n}_x + P_c B_c \hat{n}_y\right| = T^g\left|\Lambda\right|T^d \tag{8}$$

The components of the diagonal matrix Λ are the eigenvalues of the coefficient of the term $(Q_R - Q_L)$ in Equation (6), i.e.

$$\lambda_{1,2} = \vec{u}\cdot\hat{n} \tag{9}$$

$$\lambda_{3,4} = \frac{1}{2}\left[(1+\beta^2)\lambda_{1,2} \pm \sqrt{[(1-\beta^2)\lambda_{1,2}]^2 + [2\beta a\,\|\hat{n}\|]^2}\right] \tag{10}$$

The corresponding diagonalizing matrices are

$$T^g = \begin{bmatrix} 1 & 0 & \frac{1}{2\beta^2 a^2} & \frac{1}{2\beta^2 a^2} \\ u & \hat{n}_y & \frac{u+\hat{r}\hat{n}_x}{2\beta^2 a^2} & \frac{u+\hat{s}\hat{n}_x}{2\beta^2 a^2} \\ v & -\hat{n}_x & \frac{v+\hat{r}\hat{n}_y}{2\beta^2 a^2} & \frac{v+\hat{s}\hat{n}_y}{2\beta^2 a^2} \\ \frac{q^2}{2} & u\hat{n}_y - v\hat{n}_x & \frac{H+\hat{r}\lambda_1}{2\beta^2 a^2} & \frac{H+\hat{s}\lambda_1}{2\beta^2 a^2} \end{bmatrix} \tag{11}$$

$$T^d = \begin{bmatrix} 1-\dfrac{\gamma-1}{a^2}\dfrac{q^2}{2} & \dfrac{\gamma-1}{a^2}u & \dfrac{\gamma-1}{a^2}v & -\dfrac{\gamma-1}{a^2} \\ v\hat{n}_x - u\hat{n}_y & \hat{n}_y & -\hat{n}_x & 0 \\ \dfrac{\hat{s}\dfrac{q^2}{2}(\gamma-1)+\beta^2a^2\lambda_1}{\hat{t}} & -\dfrac{\hat{s}u(\gamma-1)+\beta^2a^2\hat{n}_x}{\hat{t}} & -\dfrac{\hat{s}v(\gamma-1)+\beta^2a^2\hat{n}_y}{\hat{t}} & \dfrac{\hat{s}(\gamma-1)}{\hat{t}} \\ -\dfrac{\hat{s}\dfrac{q^2}{2}(\gamma-1)+\beta^2a^2\lambda_1}{\hat{t}} & \dfrac{\hat{r}u(\gamma-1)+\beta^2a^2\hat{n}_x}{\hat{t}} & \dfrac{\hat{r}v(\gamma-1)+\beta^2a^2\hat{n}_y}{\hat{t}} & -\dfrac{\hat{r}(\gamma-1)}{\hat{t}} \end{bmatrix} \tag{12}$$

with

$$r = \lambda_3 - \lambda_1\beta^2, \qquad s = \lambda_4 - \lambda_1\beta^2, \qquad t = \frac{\lambda_4 - \lambda_3}{2} \tag{13}$$

for any stressed variable x

$$\hat{x} = \frac{x}{\sqrt{\hat{n}_x^2 + \hat{n}_y^2}}$$

and $a^2 = p\gamma/\rho$. The preconditioning involves a parameter β, which may be chosen as follows [Viozat C. Personal communication, April 1997]:

- can be chosen to be constant and equal to 1;
- can be equal to the maximum Mach number in the flow;
- can be equal to the local Mach number evaluated at Roe-averaged values:

$$M_{\text{average}} = \frac{\sqrt{u^2_{\text{average}} + v^2_{\text{average}}}}{a_{\text{average}}}$$

  where u and v are the horizontal and vertical components of the velocity respectively and a is the speed of sound;
- can be equal to the maximum of the local Mach numbers computed in neighbouring segments.

It has to be noted that for variable β, the preconditioned diagonalizing matrices discussed previously are singular at the stagnation points; thus some smoothing procedure has to be applied to the variable β.

3.2. The acoustic damping method

A modified version of the numerical flux function (6) based on a distinction of the material wave and acoustic wave part is investigated. Let δ be the acoustic damping parameter. Then, the numerical flux function can be written as follows [5,12]:

$$\Phi_F(Q_L, Q_R, \hat{n}) = \frac{F(Q_L, \hat{n}) + F(Q_R, \hat{n})}{2} - \frac{1}{2}\tilde{K}^{\delta}\left|A_c\hat{n}_x + B_c\hat{n}_y\right|(Q_R - Q_L) \tag{14}$$

where $\tilde{K}^{\delta}$ is

$$\tilde{K}^{\delta} = T^g \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \delta & 0 \\ 0 & 0 & 0 & \delta \end{bmatrix} T^d \tag{15}$$

$$\left|A_c\hat{n}_x + B_c\hat{n}_y\right| = T^g|\Lambda|T^d \tag{16}$$

Here T^g and T^d are obtained from Equations (11) and (12) for $\beta = 1$. Moreover, $T^d \cdot T^g =$ Identity. Then, one obtains from Equation (14)

$$\tilde{K}^{\delta}\left|A_c\hat{n}_x + B_c\hat{n}_y\right| = \left(T^g \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & |\delta| & 0 \\ 0 & 0 & 0 & |\delta| \end{bmatrix} |\Lambda|T^d \right) \tag{17}$$

In this study, in order not to change the dissipative properties of the flow, the acoustic damping parameter δ has to be chosen properly. However, there is no systematic way to choose the damping parameter δ.

The parameter used in the acoustic damping method affects the speed of waves associated with acoustics only; like the preconditioning algorithm proposed by Turkel [9] for incompressible flows, e.g. the third and fourth columns of T^g given in Equation (12) as well as the third and fourth rows of T^d given in Equation (11). The aim of such a process is to append a small amount of upwinding to the waves that are associated with acoustics. This results in a better accuracy in the low Mach number limit of compressible Euler equations.

The coefficient of the difference $(Q_R - Q_L)$ in both numerical flux functions given in Equations (6) and (14) are evaluated at Roe-averaged values in $\partial C_i \cap \partial C_j$ [13]

$$\tilde{\rho} = \sqrt{\rho_L \rho_R} \tag{18}$$

$$\tilde{u} = \frac{u_L + u_R\sqrt{\rho_L/\rho_R}}{1 + \sqrt{\rho_L/\rho_R}} \tag{19}$$

$$\tilde{v} = \frac{v_L + v_R\sqrt{\rho_L/\rho_R}}{1 + \sqrt{\rho_L/\rho_R}} \tag{20}$$

$$\tilde{h} = \frac{h_L + h_R\sqrt{\rho_L/\rho_R}}{1 + \sqrt{\rho_L/\rho_R}} \tag{21}$$

$$\tilde{a}^2 = (\gamma - 1)\left(\tilde{h} - \frac{\tilde{u}^2 + \tilde{v}^2}{2}\right) \tag{22}$$

They can be evaluated at the arithmetic average values in $\partial C_i \cap \partial C_j$ as it has been proved by Harten–Lax–Leer [14] for incompressible flow equations. However, there is not much difference observed between the results obtained by the numerical experiments.

3.3. MUSCL interpolation

The numerical flux functions evaluated at non-extrapolated conservative variables at the nodes will lead to first-order accuracy in space. Following the monotone upstream centred scheme for conservation laws (MUSCL) interpolation of van Leer [15], one way to reach second-order spatial accuracy is to evaluate the fluxes with extrapolated values Q_L and Q_R at the control volume face $\partial C_i \cap \partial C_j$. By definition, a difference scheme in conservation is said to be an upwind scheme if Q_L and Q_R are nearby states and Φ_{ij} is a linear approximation to the numerical flux (5). Considering Figure 1, the MUSCL interpolation of the conservative variables can be expressed as follows:

$$Q_L = Q_i + \frac{1}{2}[(1-\kappa)(\Delta_-)_i + \kappa(\Delta_+)_i] \tag{23}$$

$$Q_R = Q_j + \frac{1}{2}[(1-\kappa)(\Delta_-)_j + \kappa(\Delta_+)_j] \tag{24}$$

$$(\Delta_-)_i = Q_{x_i}(x_j - x_i) + Q_{y_i}(y_j - y_i) \tag{25}$$

$$(\Delta_+)_i = Q_j - Q_i \tag{26}$$

$$(\Delta_-)_j = Q_{x_j}(x_i - x_j) + Q_{y_j}(y_i - y_j) \tag{27}$$

$$(\Delta_+)_j = Q_i - Q_j \tag{28}$$

Here κ is the upwinding parameter and is equal to 0.5. As it is seen from the interpolation functions, none of the flux limiters are introduced into the interpolation. In spite of this, the results obtained from this kind of interpolation did not show oscillations near shocks, even for high-Mach number flows. For high-Mach number applications see Sabanca [12]. Q_{x_i}, Q_{y_i}, Q_{x_j}, Q_{y_j} in Equations (25)–(28) are computed by applying Green's theorem and taking the value of Q along an edge of the control volume to be the average of the end point values. For instance, the contribution of Q_{x_i}, along edge j–k in Figure 1 is given by

$$Q_{x_i} = \frac{1}{2A_i}(Q_j + Q_k)(y_k - y_j) \tag{29}$$

3.4. Data structure

In principle, the cells and nodes are ordered randomly for unstructured grids. Thus, the use of unstructured grids requires the storage of connectivity information along with the use of an indirect addressing system in which the nodes cannot be represented by their indices. Since numerical fluxes are to be calculated across each edge of the mesh for the nodal scheme, a data structure based on the mesh nodes can be employed. To define a node-based structure, two-integer addressing must be stored for each edge. Thus, an array must be dimensioned NEDGE(ND,2), where ND is the total number of nodes in the mesh. For each edge, the first two values in NEDGE, *SI* and *SJ*, correspond to the address of the nodes on either end of that edge, as shown in Figure 2. The addresses of the remaining vertices of each triangle on either side of that edge are also addresses of the nodes associated with the two control volumes delimited by that edge. With this information, a complete flux balance over the entire flow field can be accomplished with a simple loop over the edges. For example, for the nodal scheme, a simple loop is constructed as

```
  DO 10 I=1,ND
    SI=NEDGE(I,1)
    SJ=NEDGE(I,2)
    FLUX=FCTN(variables at N1 and N2)
    Residual(SI)=Residual(SI)+FLUX
10  Residual(SJ)=Residual(SJ)-FLUX
```

The flux at each edge is calculated only once and then added to and subtracted from the two control volumes on either side of that edge, thus ensuring conservation. The above loop will not readily vectorize due to recurrences in the last two statements. This arises from the fact that the individual control volumes (i.e. *SI* and *SJ* in this case) receive contributions from more than one edge in the mesh. This problem can be discarded by reordering the edges and grouping them into several distinct groups, such that, within each group, no control volume of the mesh is accessed more than once. This, on the other hand, results in a reduction of the vector lengths that can be used in a flux balance calculation. In the present study, these edges are grouped according to whether they are wall boundary edges or far field boundary edges or inner field edges.

3.5. Boundary conditions

At a solid wall the following treatment of boundary conditions has been chosen. In the approximation of flux balance, one row of auxiliary cells is used at the wall and the state at each mirror-image node is computed by imposing isentropic simple radial equilibrium (ISRE) at each node of the solid wall boundary. The normal velocity components at the mirror points have to be chosen so as to satisfy the slip condition after the computation of the numerical flux

function. Therefore, in a simplified ISRE approach, the normal velocity components at the auxiliary nodes are first mirrored with respect to the wall, which allows only the flux of the pressure terms to contribute to the momentum equations and thus satisfy the condition of zero mass flux and energy flux through the wall surface, see Ransbeeck and Hirsch [16].

At the inflow and outflow boundaries, a precise set of compatible exterior data, which depend on the flow regime and the velocity direction, is to be specified. Following Steger and Warming [17], fluxes are computed by using upstream values of the entropy deviation

$$S = \left(\frac{p}{p_\infty}\right)\left(\frac{\rho_\infty}{\rho}\right)^\gamma - 1 \tag{30}$$

Downstream density is computed from this relation provided that entropy deviation is zero. The integration in Equation (31) corresponds to inflow and outflow boundaries. Let Q_i be the conservative variable at the interior node and let Q_∞ be the free stream conservative variables kept at the ghost nodes. A flux vector splitting scheme following Steger and Warming [17] is applied between exterior data and interior values, i.e.

$$\oint_{\partial C_i \cap \Gamma_\infty} \vec{F}(Q)\cdot\hat{n}_i \, \mathrm{d}\sigma = \tilde{A}^+(Q_i, \hat{n}_{i\infty})\cdot Q_i + \tilde{A}^-(Q_i, \hat{n}_{i\infty})\cdot Q_\infty \tag{31}$$

where

$$\tilde{A} = A_c\hat{n}_x + B_c\hat{n}_y \tag{32}$$

and

$$\tilde{A}^+ = \tilde{L}\Lambda^+\tilde{L}^{-1}, \qquad \tilde{A}^- = \tilde{L}\Lambda^-\tilde{L}^{-1} \tag{33}$$

where Λ^+ has only positive eigenvalues, Λ^- only negative eigenvalues and such that

$$\Lambda = \Lambda^+ + \Lambda^-, \qquad |\Lambda| = \Lambda^+ - \Lambda^- \tag{34}$$

$$\tilde{A} = \tilde{A}^+ + \tilde{A}^- \tag{35}$$

$$|\tilde{A}| = \tilde{A}^+ - \tilde{A}^- \tag{36}$$

The diagonalizing matrices are given as follows:

$$\tilde{L} = \begin{bmatrix} 1 & 0 & \frac{1}{a} & \frac{1}{a} \\ u & \hat{n}_y & \frac{1}{a}(u + a\hat{n}_x) & \frac{1}{a}(u - a\hat{n}_x) \\ v & -\hat{n}_x & \frac{1}{a}(v + a\hat{n}_y) & \frac{1}{a}(v - a\hat{n}_y) \\ \frac{q^2}{2} & (u\hat{n}_y - v\hat{n}_x) & \frac{1}{a}(h + au_n) & \frac{1}{a}(h - au_n) \end{bmatrix} \tag{37}$$

$$\tilde{L}^{-1} = \begin{bmatrix} 1 - \frac{(\gamma-1)q^2}{2a^2} & (\gamma-1)\frac{u}{a^2} & (\gamma-1)\frac{v}{a^2} & -\frac{\gamma-1}{a^2} \\ (v\hat{n}_x - u\hat{n}_y) & \hat{n}_y & -\hat{n}_x & 0 \\ \frac{a}{2}\left(\frac{(\gamma-1)q^2}{2a^2} - \frac{u_n}{a}\right) & \frac{1}{2}\left(\hat{n}_x - (\gamma-1)\frac{u}{a}\right) & \frac{1}{2}\left(\hat{n}_y - (\gamma-1)\frac{v}{a}\right) & \frac{\gamma-1}{2a} \\ \frac{a}{2}\left(\frac{(\gamma-1)q^2}{2a^2} + \frac{u_n}{a}\right) & -\frac{1}{2}\left(\hat{n}_x + (\gamma-1)\frac{u}{a}\right) & -\frac{1}{2}\left(\hat{n}_y + (\gamma-1)\frac{v}{a}\right) & \frac{\gamma-1}{2a} \end{bmatrix} \tag{38}$$

This non-reflective-type boundary condition implementation allows smaller domains to be used, as there is no reflection from the outer boundary that affects the convergence of the solution.

3.6. Time integration

A semi-discrete form of the governing equations is

$$A_{\text{cell}} \frac{\partial Q_i}{\partial t} + R_i = 0 \tag{39}$$

where

$$R_i = \sum_{j \in K(i)} \Phi_{F_{ij}} \Delta s_{ij} \tag{40}$$

Temporal discretization of the governing equations is performed by using fourth-order Runge–Kutta explicit time stepping developed by Jameson [18]. The Courant condition for unstructured triangulations is

$$\frac{\Delta t}{A_{\text{cell}}} = \frac{v}{\max_{\text{faces}}(u_n + a)\Delta s_{ij}} \tag{41}$$

Here v is the Courant–Friedrich–Lewy (CFL) number. Then, time integration steps are

$$Q_i^0 = Q_i^n \tag{42}$$

$$Q_i^1 = Q_i^0 - \alpha_1 \frac{\Delta t}{A_{\text{cell}}} R_i(Q_i^0) \tag{43}$$

$$Q_i^2 = Q_i^0 - \alpha_2 \frac{\Delta t}{A_{\text{cell}}} R_i(Q_i^1) \tag{44}$$

$$Q_i^3 = Q_i^0 - \alpha_3 \frac{\Delta t}{A_{\text{cell}}} R_i(Q_i^2) \tag{45}$$

$$Q_i^4 = Q_i^0 - \alpha_4 \frac{\Delta t}{A_{\text{cell}}} R_i(Q_i^3) \tag{46}$$

$$U_i^{n+1} = U_i^4 \tag{47}$$

with

$$\alpha_1 = 0.0833, \quad \alpha_2 = 0.2069, \quad \alpha_3 = 0.4265, \quad \alpha_4 = 1.0 \tag{48}$$

To accelerate the convergence in a simple way, local time stepping could be utilized as enthalpy damping cannot be used for flux difference splitting schemes.

4. RESULTS AND DISCUSSION

The numerical methods described in the previous sections were utilized for the simulation of a variety of flow problems. Computed results indicate that both schemes are accurate, reliable and robust. On the other hand, with decreasing Mach numbers, the allowable Courant number decreases if the preconditioning parameter is variable. For this reason, the time required to reach steady state conditions increases at low subsonic Mach numbers. Moreover, once the preconditioning parameter is chosen to be constant (which must be the natural choice for high Mach numbers) numerical experiments show that the CFL number cannot exceed 1.6. On the other hand, the allowable CFL number when using the acoustic damping method is 2. In order to accelerate the convergence to steady state conditions, local time stepping can be utilized. Another simple way to accelerate the convergence is enthalpy damping, which was not used in the present study since it is inappropriate when upwind differencing is utilized [19]. Some numerical results are given here to demonstrate the capability and efficiency of the present unstructured Euler solver SUN. The unstructured grids for all the geometries within this

numerical study were generated using the programs MESH2D and MESHO provided by Dr K. Karamete. Details of these programs can be found in Reference [20].

The present code was tested on both NLR7301 (two-element) airfoil and NACA0012 airfoil for the cases shown in Table I.

Results of the simulation are compared with experimental results performed by Van Den Berg and Gooden, see Reference [21]. These conditions have been chosen because here no separation occurs, which would invalidate the Euler equations used in the present analysis. Test data are given for a two-dimensional flow past the wing flap configuration at $Re = 2.51 \times 10^6$ and a Mach number $M = 0.185$, which has been so designed that no flow separation occurs, apart from a small laminar separation bubble on the wing nose. The trailing edge flap is deflected 20°. The gap between the main airfoil element and flap element is 2.6 per cent (the ratio of the vertical distance between the main element and flap element to total chord). The experimental and computational results obtained by the present code for the pressure coefficient distributions over the main element and the trailing edge flap are shown in Figures 3 and 7 for CASE2 and CASE3 respectively. In these figures, the differences between experimental and computational results in flap configurations are due to thick boundary layer flow on the upper surface of the flaps. The convergence history is given in Figure 4 for CASE2. A detail

Table I. Test cases for the presented algorithms.

	Wall	Mach number	Angle of attack	Number of nodes	Number of triangles
CASE1	NACA0012	$M_\infty = 0.1$	0°	2242	4312
CASE2	NLR7301	$M_\infty = 0.185$	10.1°	3079	5904
CASE3	NLR7301	$M_\infty = 0.185$	6°	3079	5904

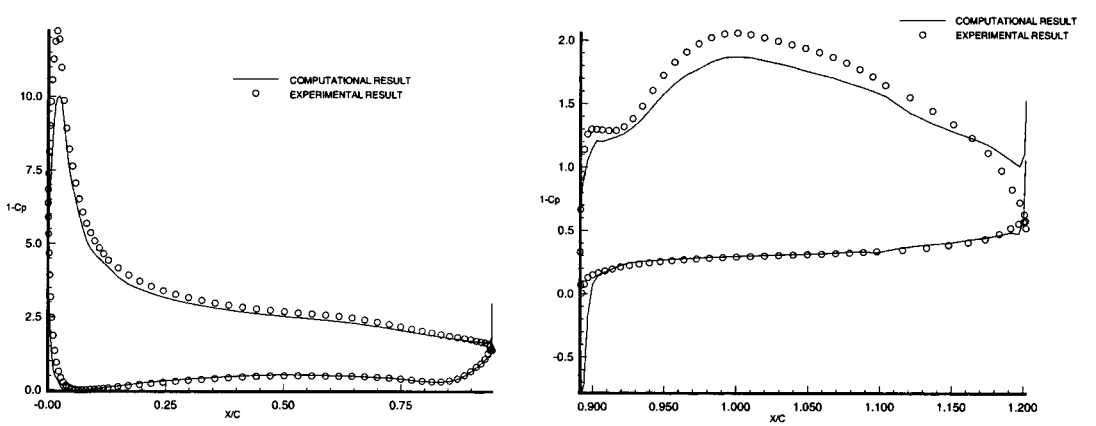

Figure 3. Comparison of Cp distribution for NLR7301 airfoil main element and flap element respectively, CASE2, preconditioned.

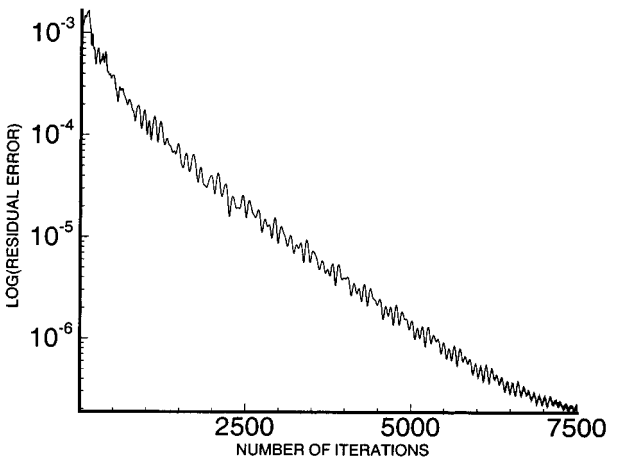

Figure 4. Convergence history for NLR7301 airfoil CASE3, $\delta = 0.5$.

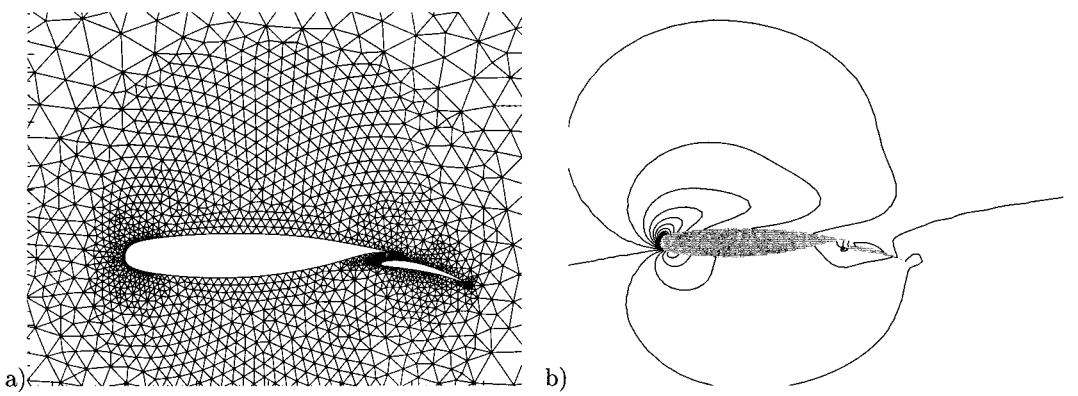

Figure 5. Enlarged view of the computational mesh for NLR7301 airfoil (a) and Mach contours for CASE2 (b), preconditioned.

of the mesh used in the computation is shown in Figure 5 and the corresponding Mach number distributions are given in Figures 5 and 6 for CASE2 and CASE3 respectively.

The preconditioning algorithm is taken as a reference point in order to investigate the effect of parameters used in the acoustic damping method. For this reason, CASE1 is divided into sub-cases as shown in Table II.

One has to be careful in choosing the acoustic damping parameter δ in order not to change the dissipative properties of the flow. A comparison of Mach contours and convergence histories is shown in Figures 8 and 9. According to Figure 8(b) and (e), the optimum δ is 1 for

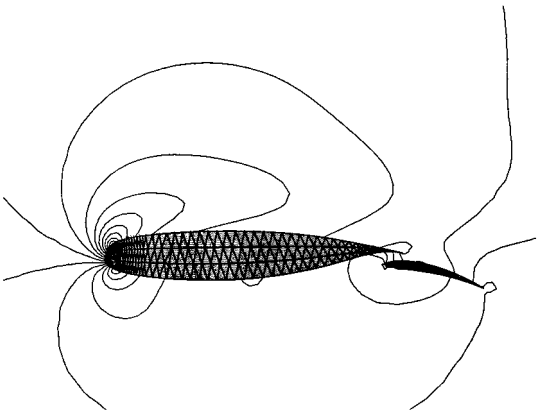

Figure 6. Mach contours for CASE3.

Table II. Comparison of lift and drag coefficients with the experimental results.

	Computational		Experimental	
	Lift	Drag	Lift	Drag
CASE2	2.7950	0.16607	2.877	0.0323
CASE3	2.229	0.240	2.416	0.0229

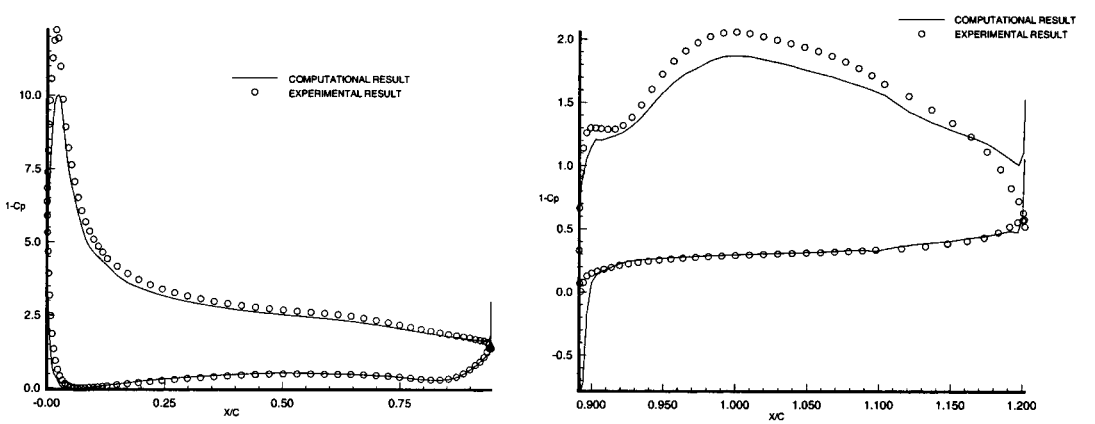

Figure 7. Comparison of Cp distributions of main element and flap element respectively with the experimental results for CASE3, $\delta = 0.5$.

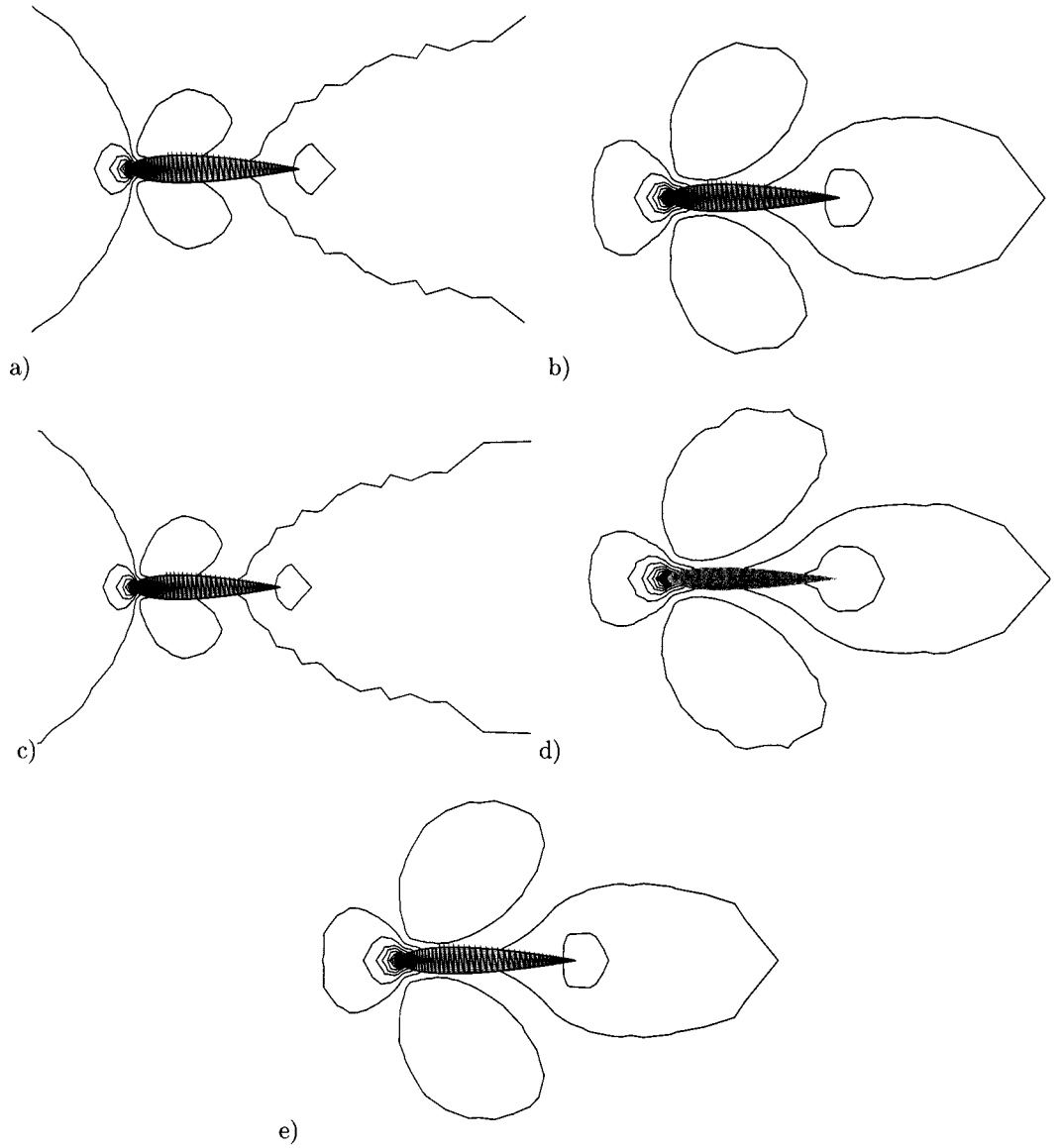

Figure 8. Mach contours for (a) CASE11, (b) CASE12, (c) CASE13, (d) CASE14 and (e) CASE15.

CASE1. By the numerical experiments, this optimum acoustic damping parameter is changing by the free stream Mach number and the maximum allowable δ is 2 for CASE1. Considering CASE11 and CASE13 in Tables III and V, the flow is dissipative in regions where the flow is

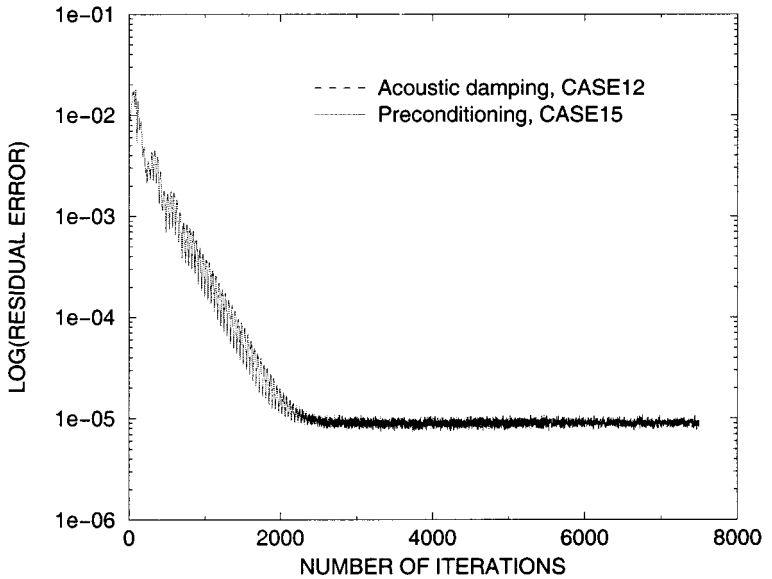

Figure 9. Convergence histories for CASE12 and CASE15.

Table III. Parameters used for CASE1 in the presented solutions.

	Method	Parameters used
CASE11	Acoustic damping	0.5
CASE12	Acoustic damping	1.0
CASE13	Acoustic damping	$\frac{1}{6}$
CASE14	Acoustic damping	$\frac{3}{2}$
CASE15	Preconditioning variable	β

Table IV. Computational cost for the low-Mach number flow simulations.

Case number	CPU time (s)
CASE14	1567.40
CASE15	3361.60

expected to be viscous when δ is small as is shown in Figure 8(a) and 8(c). On the other hand, when δ is chosen to be greater than its optimum value, CASE14, the solution shows good behaviour in the near wall regions but becomes dissipative in uniform regions of the flow as is shown in Figure 8(c). For the optimum value of δ, i.e. CASE12, the obtained solution dictates the solution obtained by preconditioning, CASE15, as is seen from the maximum and the minimum Mach numbers given in Table V and from Figure 8(b) and (e). The advantage of the acoustic damping method compared with preconditioning is its computational

Table V. Comparison of maximum and minimum Mach numbers obtained in NACA0012 airfoil flow simulations.

	M_{max}	M_{min}
CASE11	0.1127	0.009128
CASE12	0.1086	0.01854
CASE13	0.1127	0.0091308
CASE14	0.17051	0.0244405
CASE15	0.1084	0.018412

efficiency, as can be seen from Table IV, where the CPU time obtained on an IBM/ + AIX RS6000 are compared. The convergence histories for the optimum acoustic damping parameter for CASE12 and the preconditioning algorithm CASE15 are compared in Figure 9.

5. CONCLUSION

We have investigated for the two-dimensional unsteady, compressible Euler equations, the capability to accurately calculate low-Mach number flows, which are already assumed to be incompressible in this limit. Similar work has already been performed by Viozat [8,11], using an explicit preconditioning algorithm. It was found that a suitably modified acoustic damping algorithm investigated in the present study will give the same results. The advantage of the modified acoustic damping algorithm is that the implementation is easier and the CFL number does not degrade with decreasing Mach number compared with a preconditioning algorithm. This allows faster convergence to steady state calculations and the error is small enough in some engineering sense. The present code shows good behaviour in solving moderately low-Mach number flows. Moreover, the obtained solutions are fourth-order time and second-order space accurate. However, solutions at very low Mach numbers with this approach showed that the accumulations of round-off errors in pressure terms affect the convergence and the accuracy of the solutions by the numerical experiments.

Future work will concentrate on the development of more general ideas to increase the performance of the solver at low Mach numbers and to increase the computational efficiency by parallelizing and by implementing solution adaptive procedures.

ACKNOWLEDGMENTS

The authors would like to express their deepest gratitudes to C. Viozat for her fruitful discussions in the development of this study and Dr K. Karamete for his technical assistance in grid generation, as well as Dr A. Melling for the corrections in the text.

REFERENCES

1. Karki KC, Patankar SV. Pressure based calculation procedure for viscous flows at all speeds in arbitrary configurations. *AIAA Journal* 1989; **27**(9): 1167–1174.

2. Lilek Z. Ein Finite-Volumen Verfahren zur Berechnung von inkompressiblen und kompressiblen Strömungen in komplexen Geometrien mit beweglichen Rändern und freien Oberflächen. PhD thesis, Institut für Schiffbau, Universität Hamburg, 1995.
3. Klein R. Semi-implicit extension of a Godunov-type scheme based on low Mach number asymptotics I: one-dimensional flow. *Journal of Computational Physics* 1995; **121**: 213–237.
4. Munz CD, Klein R. The multiple pressure variable approach for the numerical approximation of weakly compressible fluid flow. In *Proceedings of the 2nd International Conference on Numerical Methods for Fluid Flows*, Prague, June, Feistauer M, Rannacher R, Kozel K (eds), 1994.
5. Beux F, Lanteri S, Dervieux A, Larrouturou B. Upwind stabilization of Navier–Stokes solvers. INRIA, Rapport de Recherche, No. 1885, March, 1993.
6. Choi D, Merkle CL. Application of time-iterative schemes to incompressible flow. *AIAA Journal* 1985; **23**: 1518–1524.
7. Choi YH, Merkle CL. The application of preconditioning in viscous flows. *Journal of Computational Physics* 1993; **105**: 207–223.
8. Guillard H, Viozat C. On the behaviour of upwind schemes in the low Mach number limit. INRIA, Rapport de Recherche, No. 3160, April, 1997.
9. Turkel E. Preconditioned methods for solving the incompressible and low speed compressible equations. *Journal of Computational Physics* 1987; **72**: 277–298.
10. Volpe G. Performance of compressible flow codes at low Mach numbers. *AIAA Journal* 1993; **31**(1): 49.
11. Viozat C. Implicit upwind schemes for low Mach number compressible flows. INRIA, Rapport de Recherche, No. 3084, January, 1997.
12. Sabanca M. Development of a two-dimensional Euler solver using finite volume method for external flows. MSc thesis, Middle East Technical University, 1997.
13. Roe PL. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics* 1981; **43**: 357–372.
14. Harten A, Lax PD, van Leer B. On upstream differencing and Godunov type schemes for hyperbolic conservation laws. *SIAM Review* 1983; **25**(1): 35.
15. van Leer B. Towards the ultimate conservative difference scheme V: a second-order sequel to Godunov's method. *Journal of Computational Physics* 1979; **32**: 101–136.
16. Van Ransbeeck P, Hirsch C. Multidimensional upwind schemes for the Euler/Navier–Stokes equations on structured grids. *Notes on Numerical Fluid Mechanics* 1997; **57**: 305–338.
17. Steger JL, Warming RF. Flux vector splitting of the inviscid gas dynamic equations with application to finite-difference methods. *Journal of Computational Physics* 1981; **40**: 263–293.
18. Jameson A, Schmidt W, Turkel E. Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time stepping schemes. *AIAA Journal* 1981; **81**: 1259.
19. Batina JT. Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA Journal* 1990; **28**: 1381.
20. Karamete K. A general unstructured mesh generation algorithm with its use in CFD applications. PhD thesis, Middle East Technical University, 1996.
21. Anon. A selection of experimental test cases for the validation of CFD codes. AGARD, Advisory Report No. 303, I, II, 1994.

